A NEW APPROACH TO TEXT UNDERSTANDING

Ralph Weischedel, Damaris Ayuso, Sean Boisen, Heidi Fox, Robert Ingria

BBN Systems and Technologies
10 Moulton St.
Cambridge, MA 02138

ABSTRACT

This paper first briefly describes the architecture of PLUM, BBN's text processing system, and then reports on some experiments evaluating the effectiveness of the design at the component level. Three features are unusual in PLUM's architecture: a domain-independent deterministic parser, processing of (the resulting) fragments at the semantic and discourse level, and probabilistic models.

1. INTRODUCTION

The PLUM (Probabilistic Language Understanding Model) natural language understanding system for extracting data from text is based on three unusual features: probabilistic language models, a domain-independent deterministic parser, and processing of (the resulting) fragments at the semantic and discourse level. Earlier papers have focused on the probabilistic aspects of the system [Weischedel et al., 1991; de Marcken, 1990]; here we focus on the other two design features.

While several deterministic parsers have been constructed based on Marcus's Determinism Hypothesis, PLUM seems to be the first application system that employs a deterministic parser. Many systems have been built based on semantic and discourse-level processing of fragments, most notably systems based on conceptual dependency and scripts [Schank and Riesbeck, 1981]. However, PLUM may be the first system that uses a hybrid of such semantic techniques with the purely syntactic processing of Marcus's Determinism Hypothesis, two approaches that seemed totally antithetical when first proposed.

The impact of marrying those two techniques is a robust system that produces answers in the application domain in spite of syntactic complexity, syntactic ill-formedness, extra-grammaticality and a high percentage of unknown words. A second impact is that the system can produce answers at a very early stage of porting it to a new domain. Both of these claims are substantiated in this paper by evaluating the performance of the system as the lexicon grows, without changing the syntactic, semantic, and discourse rules of the system.

2. BRIEF SYNOPSIS OF SYSTEM COMPONENTS

Major system components are shown in the diagram in Figure 1. We expect the particular implementations to change and improve substantially during the next two years of research and development. A preprocessor driven by finite state rules divides the message into header material (if any), paragraphs, sentences, and trailer material (if any).

A well-known problem in using deterministic parsing is the fact that most words in English are ambiguous even regarding part of speech. In the Foreign Broadcast Information Service texts of MUC-3, we estimate that the vocabulary had an average ambiguity of over two parts of speech in the TREEBANK tag system.

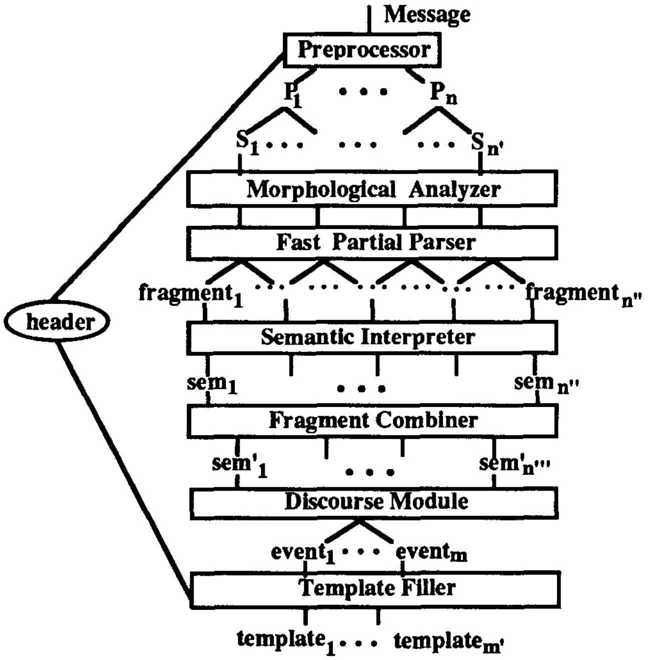
Message
Preprocessor
P_1 • • • P_n
S_1 • • • • • • • • • $S_{n'}$
Morphological Analyzer
Fast Partial Parser
header
$fragment_1$ • • • $fragment_{n''}$
Semantic Interpreter
sem_1 • • • $sem_{n''}$
Fragment Combiner
sem'_1 • • • $sem'_{n'''}$
Discourse Module
$event_1$ • • • $event_m$
Template Filler
$template_1$ • • • $template_{m'}$

Figure 1. PLUM System Architecture

In PLUM, determining the part of speech of highly ambiguous words is performed by well-known Markov modelling techniques. Though part of speech ambiguity was high, the only ambiguity that negatively impacted performance in extracting the desired information from text was recognizing proper nouns, since the text is upper case only, and the set of names is open-ended, as is the general vocabulary. To improve the recognition of Latin American names, we employed a statistically derived five-gram (five letter) model of words of Spanish origin and a similar five-gram model of English words, under the assumption that

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE **1992**	2. REPORT TYPE	3. DATES COVERED **00-00-1992 to 00-00-1992**

4. TITLE AND SUBTITLE **A New Approach to Text Understanding**	5a. CONTRACT NUMBER
	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **BBN Technologies,10 Moulton Street,Cambridge,MA,02238**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR’S ACRONYM(S)
	11. SPONSOR/MONITOR’S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES

14. ABSTRACT

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**		**7**	

words of Spanish origin in these English texts about Latin America were probably proper names.

The parser and grammar are designed to find analyses for a non-overlapping sequence of fragments, as represented by the multiple fragments in Figure 1. When cases of permanent, predictable ambiguity arise, such as a prepositional phrase that can be attached in multiple ways, most conjoined phrases, commas, and parentheses, the parser closes the analysis of the current fragment (including all open constituents), and begins the analysis of a new fragment.

This is a departure from Marcus's D-Theory proposal, where an ancestor relationship for each constituent must be stated even if a parent relationship cannot be. Thus, in the example below, *but* and *no injuries* are analyzed, but not placed under any node. The departure does not seem to hurt semantic processing, since the critical entities in the text and some relations between them are found in every sentence, whether syntactically ill-formed, complex, novel, or straightforward. Rather, the impact was beneficial, for we were able to produce output of the whole system much earlier than if the grammar rules, semantic rules, and lexicon had to be more complete.

Unlike the previous systems based on conceptual dependency and script application [Schank and Riesbeck, 1981], this parsing is done using domain-independent syntactic rules.

The deterministic parser employed was developed by de Marcken at MIT [de Marcken 1990]. Though we have not (yet) made substantial changes to the parsing code nor to the grammar, we are replacing his "disambiguator", which deals with part-of-speech ambiguity with our stochastic part-of-speech tagger (POST) [Meteer, et al. 1991]. The resulting syntactic component is named the Fast Partial Parser (FPP). "Here are the parse fragments generated for the sentence, “THE BOMBS CAUSED DAMAGE BUT NO INJURIES”:

```
("THE BOMBS CAUSED DAMAGE"
  (S (NP (DET "THE") (N "BOMBS"))
     (VP (AUX) (VP (V "CAUSED")
                   (NP (N "DAMAGE"))))))
("BUT" (CONJ "BUT"))
("NO INJURIES"
  (NP (DET "NO")(N "INJURIES")))
("." (PUNCT "."))
```

Each fragment is processed by the semantic interpreter, producing a partial semantic representation in a frame language, like KL-ONE. (See Figure 2.) Semantic analysis is shallow; for example, in Figure 2 the pp-modifier slot of the entity corresponding to *the embassies of the PRC* is not semantically analyzed further by the semantic interpreter, e.g., to determine whether the PRC owns the embassy buildings, whether the PRC uses the embassy buildings, etc. Shallow analysis is necessary since most of the words in an article are semantically unknown, and since it is highly desirable that some analysis be produced for each fragment to avoid totally missing information. (Jacobs et al. [1991] estimate that 75% of the words in these MUC texts are not relevant.)

The semantic interpreter uses structural rules; nearly all of these carry over to all new domains. Domain-dependent, lexical semantic rules contain traditional case frame information, e.g., the logical object of a murder is a living thing. The novel aspect in PLUM is that the case frames for verbs were hypothesized by a statistical induction algorithm [Weischedel, et al., 1991a]. Each hypothesized case frame was manually reviewed over a two day period, rather than the weeks or even months of effort that might normally be involved in writing case frames for verbs. The frame-based semantic representation for an unusually short and simple sentence appears in Figure 2.

Based on local syntactic and semantic information, a fragment combining algorithm combines phrases to provide more complete analyses of the input [Weischedel, et al., 1991a]. The current set of fragment combining rules focus on finding conjoined phrases,[1] prepositional phrase attachment[2], appositive recognition, and on correcting some errors made by the parser (e.g., combining adjacent fragments into a single noun phrase). Though there was no time to integrate and test this component for use MUC-3 in May, 1991, an experiment on the improvement in syntactic analyses produced based on this component is included in this paper.

Our fragment combining code is rule-based, and can take into account syntactic categories, simple properties of the tree configuration (for example, whether a node is the only child of its parent), and semantic type. The simplest attachment strategy is to process the fragments of a sentence from left to right, considering each pair of successive fragments. For each pair of fragments, all possible attachment points on the right edge of the left fragment are considered, starting from the lowest (closest) node. Some rules consider more than one fragment to the right, for example, combining an NP with commas on each side into a single appositive NP. Therefore, as in the deterministic parser, decisions are made locally, rather than assuming global context.

[1] The parser usually produces fragments where a conjoined phrase appears because local syntactic information is typically not sufficient to reliably predict the correct parse.

[2] The parser usually does not attach prepositional phrases because of the inherent ambiguity.

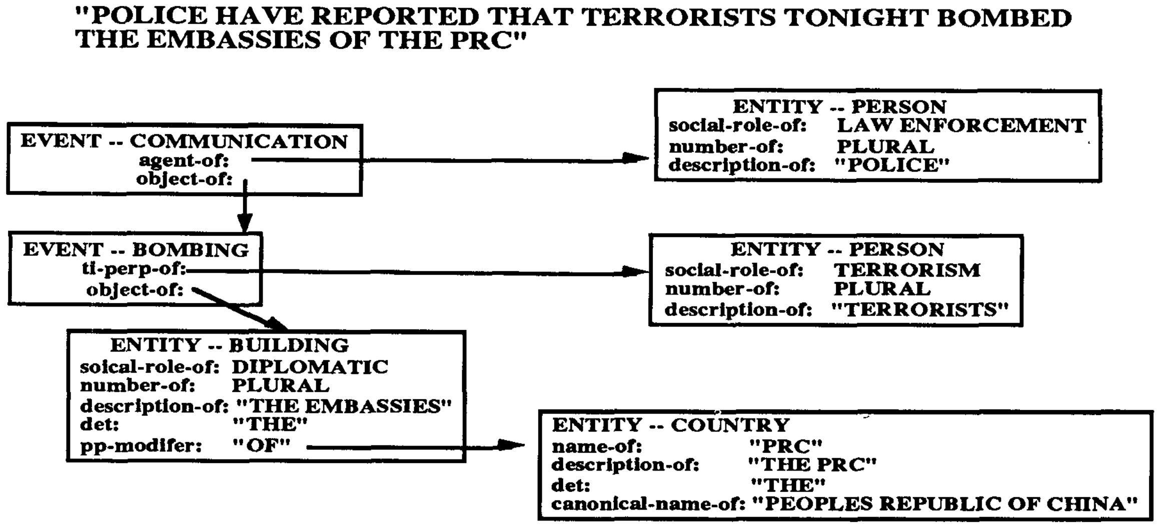
"POLICE HAVE REPORTED THAT TERRORISTS TONIGHT BOMBED THE EMBASSIES OF THE PRC"
EVENT -- COMMUNICATION
agent-of:
object-of:
ENTITY -- PERSON
social-role-of: LAW ENFORCEMENT
number-of: PLURAL
description-of: "POLICE"
EVENT -- BOMBING
ti-perp-of:
object-of:
ENTITY -- PERSON
social-role-of: TERRORISM
number-of: PLURAL
description-of: "TERRORISTS"
ENTITY -- BUILDING
soical-role-of: DIPLOMATIC
number-of: PLURAL
description-of: "THE EMBASSIES"
det: "THE"
pp-modifer: "OF"
ENTITY -- COUNTRY
name-of: "PRC"
description-of: "THE PRC"
det: "THE"
canonical-name-of: "PEOPLES REPUBLIC OF CHINA"

Figure 2: Example Semantic Representation

The discourse component performs three tasks: hypothesizing relevant events from diverse descriptions, recognizing co-reference, and hypothesizing values for components of an event. Discourse processing would normally look for entities to fill roles in stated predicates, as Hobbs [1989, 1988] has argued. Since complete syntactic accounts of a sentence are not usually found by our system,[3] semantic representations of events and states of affairs have more unfilled slots (roles) than if complete syntactic analyses were found. In our case, there are simply more such unfilled roles, and less syntactic relations helping out. A second challenge faced by the discourse component is that reference resolution must be performed with limited semantic understanding. Given these challenges, it is clear from the test results in MUC-3 that the discourse component does reconstruct event structure well, in spite of missing syntactic and semantic relations.

An example frame for an event produced by discourse processing appears in Figure 3. A score of 0 indicates the filler was found directly by the semantics; a score of 1 indicates it was in the same fragment; 2 indicates it was in the same sentence; 4 indicates it was found in the same paragraph; and 6 that it was found in an adjacent paragraph. Note that El Salvador, though not in the text, was introduced by the definition of San Isidro in the lexicon, which had only been seen previously as a town of El Salvador.

The template generator has three tasks: finding and/or merging events hypothesized by discourse processing into a complete template structure, deciding whether to default the value of template slots not found in the event structure (e.g, using date and location information in the header), and creating the required template forms.

3. EVALUATION

The system as a whole was formally evaluated in the Government-sponsored Third Message Understanding Conference (MUC-3), and scored among the top systems in extracting data from text [Proceedings of MUC-3, 1991]. In this paper we report on two additional experiments run since then to assess component contributions to the system.

3.1 Lexicon

If the grammar rules and semantic rules are both compositional and domain-independent, one would expect the recall of the system (the percent of information correctly found by the system out of all desired information in the text) to grow linearly at first as the lexicon grows followed by tapering off to an asyptote.[4]

To test this, we ran the system after randomly removing lexical entries (though not removing a word's part of speech). The results with various percentages of the lexicon and with linear curve fitting appear in Figure 4.

[3] Apparently none of the fifteen systems entered in the Third Message Understanding Conference (MUC-3) usually found complete syntactic analyses of the long, complex sentences in the MUC-3 corpus.

[4] Of course, when the lexicon is so small that very little information is found at all, recall might not increase linearly as the lexicon grows. Presumably, at some point asymptotic growth must limit as recall approaches 100%.

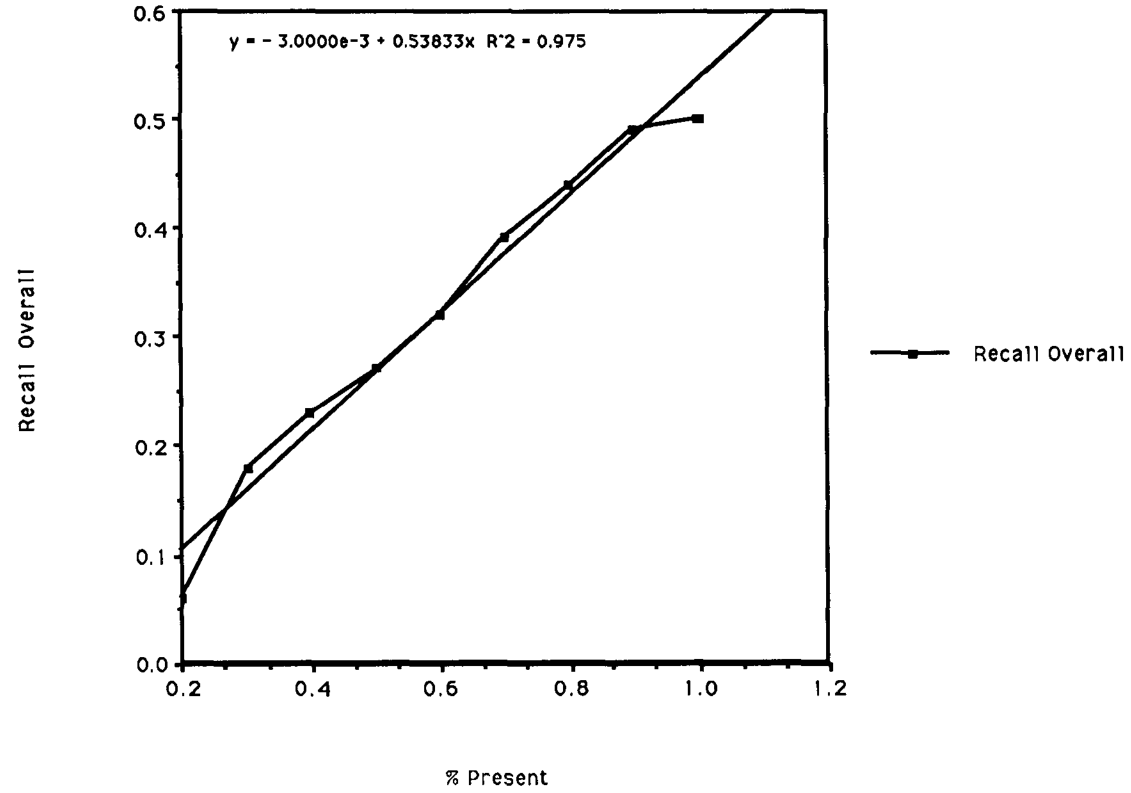

Figure 4: Growth in Ability to extract Data from Text as the Lexicon Grows

"POLICE HAVE REPORTED THAT TERRORISTS TONIGHT BOMBED THE EMBASSIES OF THE PRC AND THE SOVIET UNION. THE BOMBS CAUSED DAMAGE BUT NO INJURIES."

"A CAR-BOMB EXPLODED IN FRONT OF THE PRC EMBASSY, WHICH IS THE LIMA RESIDENTIAL DISTRICT OF SAN ISIDRO. MEANWHILE, TWO BOMBS WERE THROWN AT A USSR EMBASSY VEHICLE THAT WAS PARKED IN FRONT OF THE EMBASSY LOCATED IN ORRANTIA DISTRICT, NEAR SAN ISIDRO."

Event: BOMBING
Trigger: "BOMBED" (?29)
Slots:
TI-PERP-OF: "TERRORISTS" (?9, score=0)
EVENT-TIME-OF:
EVENT-LOCATION-OF:
"EL SALVADOR" (?100, score=6)
" SAN ISIDRO" (?104, score=6)
" RESIDENTIAL DISTRICT" (?105, score=6)
"ORRANTIA DISTRICT" (?169, score=6)
TI-INSTR-OF : "THE BOMBS" (?41, score=4)
TI-RESULT-OF:
"DAMAGE" (?46, score=4)
"NO INJURIES" (?54, score=4)
OBJECT-OF: "THE EMBASSIES" (?22, score=0)

Figure 3: An Example Event Produced

Precision, the percent of data correctly extracted out of all the information extracted, should be relatively unaffected in a compositional, domain-independent system. That is, if the lexicon is declarative rather than itself containing rules, the quality of answers produced should be unaffected. Precision in tests corresponding to the recall data plotted in

Figure 4 varied only 5% throughout the range; the difference between having only 20% of the lexicon to having the full lexicon was only 2% in precision.

3.2 Deterministic parser and grammar of English versus Fragment combining.

In the experiment reported here, only a small set of fragment combining rules were tested, those deemed to be most useful in the ability to extract information fro MUC-3; no attempt to provide coverage for the full variety of English syntax has been made. The fragment combining rules were as follows ranked by frequency of occurrence in the experiment are as follows:

- PP attachment to an NP (55%)
- PP attachment to a VP (14%)
- merging of several N's into a single NP (13%)
- combing appositive NPs (7%)
- attaching a conjoined NP (6%)
- PP attachment to an ADJP (3%)
- attaching time NP to VP (1%)
- repairing dates (< 1%)

To evaluate the relative contribution of the deterministic parser and the fragment combining component, we used recently developed grammar evaluation software [Black, et al., 1991]. This software uses TREEBANK parse trees as a reference answer. To factor out most grammatical idiosyncracies where legitimate theoretical differences may exist, a TREEBANK tree is reduced by a homomorphism to essential phrase bracketings, such as that in Figure 5. The user of the evaluation software then writes a homomorphism component that reduces his/her parser's output to a similar bracketed form. Then a comparator in the evaluation software counts three things:

- *Recall*, the number of bracketed phrases in both answers divided by the number of bracketed phrases in the reference answer
- *Precision*, the number of bracketed phrases in both answers divided by the number of bracketed phrases in the system's output
- *Crossings*, the number of times a system phrase crosses a bracketed boundary in the reference answer.

TREEBANK Tree (without parts of speech

```
 (S (NP the Catholic church)
   (X has
      (VP expressed
        (NP satisfaction
              (PP with
                (NP the investigations
                      (PP in
                        (NP the case
                              (PP of
                                (NP the murdered Jesuits))))))))
        and
        (X is
          (VP encouraging
                  (NP the government)
                  (S (NP *)
                    to
                    (VP continue
                      (S (NP *)
                        to
                          (VP search
                            (PP for
                                (NP the perpetrators
                                  (PP of
                                    (NP this crime)))))))))))))
.)
```

Reduced Form Given TREEBANK Parse Tree

```
[[THE CATHOLIC CHURCH]
         [HAS
          [EXPRESSED
           [SATISFACTION
              [WITH
                [THE INVESTIGATIONS
                    [IN
                      [THE CASE
                        [OF [THE MURDERED JESUITS]]]]]]]
           AND
           [IS
                   [ENCOURAGING
                    [THE GOVERNMENT]
                    [CONTINUE
                     [SEARCH
                         [FOR
                             [THE PERPETRATORS
                                     [OF [THIS CRIME]]]]]]]]]]
```

Reduced Form Given PLUM Parse Tree

```
[[[THE CATHOLIC CHURCH]
           [HAS [EXPRESSED SATISFACTION]]]
 [WITH [THE INVESTIGATIONS]]
 [IN
  [[THE CASE]
           [OF [THE MURDERED JESUITS]]]]
 AND
 [IS
           [ENCOURAGING
            [THE GOVERNMENT]
            [CONTINUE SEARCH]]]
 [FOR
           [[THE PERPETRATORS]
            [OF [THIS CRIME]]]]]
```

Figure 5: Parse Trees Reduced to Minimal Bracketing for Parser Evaluation

In using the evaluation software, it became readily apparent that the absolute numbers output for our deterministic parser were not particularly informative, though the relative performance change from one parser run to another was instructive. To see this, consider the example in Figure 5. The input sentence contains a prepositional phrase whose attachment is ambiguous. Therefore, the system, by design, closes the constituents up until the prepositional phrase; however, the evaluator counts this as three crossings (three errors) for the one design feature. Since permanent predictable ambiguity occurs frequently in the long, textual sentences of the MUC corpus, this multiplicative penalty is applied very often.

However, relative comparison of one parser to measure system improvement (or retrenchment) over time is valuable. For instance, on a test set of 900 sentences, our fragment combining component successfully found 1,000 more phrases than running the deterministic parser alone, eliminated 250 incorrect structures, and reduced the total number of crossings by 300.

4. CONCLUSIONS

PLUM has the following key features:

1. Deterministic parsing and semantic interpretation of all fragments produced.

2. Event-based and template-based knowledge to find relations among entities when syntax/semantics cannot find them.

3. Statistical language models at multiple levels.

These were key to PLUM performing among the top systems evaluated in MUC-3. Because of the focus on producing syntactic and semantic analyses of fragments when no complete analysis was possible, and because of the assumption that discourse processing can fit the fragments together based on required roles in defined event structures, the system can produce answers end-to-end very early on when porting to a new domain, long before domain-specific lexical items and any domain-specific semantic rules are complete.

That conclusion is supported quantitatively by our experiments. Recall, the percent of information in the text correctly extracted, grew nearly linearly as the lexicon grew in the experiment. Precision, the percent correct information extracted of information output by PLUM, remained flat as a function of lexicon size, also supporting the intuition that the lexicon was declarative and separate from rule-based processing.

A second conclusion is that deterministic parsing can be supplemented by locally applied, fragment combining rules that use both the syntactic and semantic properties of fragments produced to resolve ambiguity that syntax alone can not resolve in a deterministic parser. The experiment reported here demonstrates that.

The degree of success obtained by marrying domain-independent, deterministic parsing with partial understanding and statistical techniques has been quite gratifying. The techniques which seemed so incompatible and antithetical in the seventies have proven synergistic.

ACKNOWLEDGMENTS

The work reported here was supported in part by the Defense Advanced Research Projects Agency and was monitored by the Rome Air Development Center under Contract No. F30602-91-C-0051. The views and conclusions contained in this document are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of the Defense Advanced Research Projects Agency or the United States Government.

REFERENCES

1. Black, E., et al., A Procedure for Quantitatively Comparing the Syntactic Coverage of English Grammars, Proceedings of the Fourth DARPA Workshop on Speech and Natural Language, 1991.

2. de Marcken, C.G. Parsing the LOB Corpus. Proceedings of the 28th Annual Meeting of the Association for Computational Linguistics 1990, pp. 243-251.

3. Hobbs, J.R. Coherence and Coreference, Cognitive Science, Vol. 3, No. 1, 1979, pp. 67-90.

4. Hobbs, J.R. et. al., Interpretation as Abduction, Proceedings of the 26th Annual Meeting of the Association for Computational Linguistics, 1988, pp. 95-103.

5. Jacobs, P., Krupka, G.P., and Rau, L.F. Lexicon-Semantic Pattern Matching as a Companion to Parsing in Text Understanding, Proceedings of the Fourth DARPA Workshop on Speech and Natural Language, Morgan Kaufmann Publishers, San Mateo, CA, February 1991, pp. 337-341.

6. Meteer, M., Schwartz, R., and Weischedel, R. Empirical Studies in Part of Speech Labelling., Proceedings of the Fourth DARPA Workshop on Speech and Natural Language, Morgan Kaufmann Publishers, San Mateo, CA. February 1991, pp. 331-336.

7. Proceedings of the Third Message Understanding Conference. Morgan Kaufmann Publishers, San Mateo, CA, 1991.

8. Schank, R.C. and Riesbeck C.K., Inside Computer Understanding, Lawrence Erlbaum Associates, Inc., 1981.

9. Weischedel, R., Ayuso, D.M., Bobrow, R., Boisen, S., Ingria, R., and Palmucci, J., Partial Parsing, A Report on Work in Progress, Proceedings of the Fourth DARPA Workshop on Speech and Natural Language, Morgan Kaufmann Mateo, CA, 1991a, pp. 204-210.

10. Weischedel, R., Meteer, M., and Schwartz, Applications of Statistical Language Modelling to Natural Language Processing, unpublished manuscript, 1991b.